Evaluation of Machine Learning Models for Product Attribute Prediction Using Etsy Dataset

Joseph Mathew
Joseph.mathew8@mail.dcu.ie
School of Computing, Dublin City University
Dublin, Ireland

ABSTRACT

"Random Forest, Logistic Regression, Linear SVC, and Naive Bayes are four machine learning algorithms that are used in this study to predict important product attributes such as top category ID, bottom category ID, primary color ID, and secondary color ID from the Etsy dataset. The main goal was to find which model gives the highest F1 score, focusing on balancing the precision and recall. A deliberate feature selection process was used to improve the models' predicted accuracy. According to the results, Logistic Regression and Linear SVC consistently perform better than other models in terms of F1 score, indicating that they are appropriate for predicting categorical data in e-commerce settings".

1. INTRODUCTION

E-commerce platforms like Etsy provide a vast amount of data. Effective data analysis may greatly improve consumer happiness and product management in the ever-changing world of e-commerce. The prediction of four crucial product features on Etsy is the focus of this study: primary color ID, secondary color ID, bottom category ID, and top category ID. Precise estimation of these characteristics is essential to enable effective product classification and enhance search capabilities, thereby augmenting the user experience.

With an emphasis on F1 scores, the study evaluates the effectiveness of four machine learning algorithms: Random Forest, Logistic Regression, Linear SVM, and Naive Bayes. The F1 score is crucial in this context as it measures the balance between precision and recall. The most successful models were found to be Logistic Regression and Linear SVM, demonstrating their accuracy and resilience when dealing with categorical data. on addition to illuminating the algorithmic advantages and disadvantages, this study offers useful advice for implementing these models on e-commerce sites such as Etsy.

2. RELATED WORKS

(Fang et al., 2021) describes a multimodal learning model that was created for Shopee's product matching, with an emphasis on integrating sophisticated text and picture processing methods to improve the shopping experience. For image embeddings, it makes use of NFNet, Swin Transformer, and EfficientNet; for text embeddings, it makes use of Distil-Bert, Albert, Multilingual Bert, and TF-IDF. Using cosine similarity and distance metrics, a KNN classifier is used to integrate and classify these. The model uses an ensemble of seven models to ensure accurate predictions and makes use of the Arcface loss function to enhance training efficacy.

(Bhavani and Kumar, 2021) explains how different machine learning algorithms are applied and evaluated for text classification, which is a critical component of natural language processing because of the exponential rise of digital data. Preprocessing, feature extraction—especially using TF-IDF vectors—and the application of several classification algorithms, including K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Decision Tree, and Neural Networks (both

CNN and RNN), are all covered in the full classification workflow. The effectiveness and difficulties of each approach in managing multi-class text classification are highlighted by evaluating these techniques based on recall and accuracy. The study offers insights into the relative performance of different technologies in automated document sorting, highlighting in particular the efficacy of SVM and deep learning approaches for managing vast and heterogeneous datasets.

(Khurana, Samulowitz, and Turaga, 2018) discusses an innovative framework that uses reinforcement learning to drive a performance-driven exploration of a transformation graph in order to automate feature engineering in predictive modeling. By methodically capturing and investigating the set of possible feature transformations, the suggested approach enables the effective identification of useful features at a lower cost in terms of both compute and labor. The method maximizes feature selection and reduces exploration by evaluating transformations based on cross-validation performance inside a structured transformation network. The method shows a considerable reduction in error rates and CPU demand compared to standard methods, and has been evaluated on 48 different datasets.

The application of Support Vector Machines (SVM) and other techniques, such as Rocchio, Naive Bayes, and Logistic Regression, to text categorization on English documents is covered by (Luo, 2021). Weka is used to conduct experiments, demonstrating that SVM achieves above 90% classification rates when working with huge feature sets. The paper highlights how different machine learning techniques vary in efficacy depending on dataset size by evaluating them using precision, recall, and F1 scores. It indicates the potential IT applications of these methods, with possible R, TensorFlow, Python, or Matlab modifications.

3. METHODOLOGY

3.1 DATA PREPROCESSING

Column Transformer: We use the Column Transformer to apply various preprocessing procedures to distinct feature subsets in our dataset. This is essential for effectively managing the range of data kinds (textual and category). Because each feature is handled separately but inside a single framework, the workflow is made simpler and consistency is maintained during feature transformations.

HashingVectorizer: For machine learning models to process natural language data, preprocessing involves converting text input into a fixed-size numerical representation, which is accomplished with the help of HashingVectorizer. This converts the text into a high-dimensional sparse matrix using a hashing function, which lowers memory consumption and expedites processing without the need for fitting.

TF-IDF Transformer: After vectorization, the TF-IDF (Term Frequency-Inverse Document Frequency) Transformer is used to modify the frequency of terms based on how frequently they occur in the texts. The effect of tokens that appear often and are therefore less informative than features that occur in a smaller fraction of the training corpus is helped to be downscaled.

OneHotEncoder: Categorical data is processed with OneHotEncoder. It transfers the category feature or features to a binary numerical representation without adding any ordinality. In order to make categorical data easier to handle for machine learning algorithms that need numerical input, each unique category value is converted into a one-hot vector.

3.2 MODELS USED

I looked at a wide range of machine learning models in my first research to see which would work best for our classification challenge. Among them were:

Random Forest: A reliable ensemble method that makes predictions by utilizing several decision trees. It was finally decided against using it for the final implementation because of its relatively slower training times and difficulty in adjusting for large datasets, despite its excellent accuracy and capacity to represent complex interactions in data.

Support vector classifier (SVC) model: This model is effective for high-dimensional spaces since it is based on the idea that decision boundaries are defined by decision planes. Although linear SVC works well in binary classification situations, it takes a long time to compute, particularly when dealing with huge datasets and multi-class environments.

The final selection of **Naive Bayes** and **Logistic Regression** was driven by multiple factors:

Accuracy: In early testing, both models showed good classification accuracy. A strong balance between interpretability and accuracy was offered by logistic regression, which was particularly helpful in situations where it was essential to comprehend the influence of several parameters. However, despite assuming feature independence, Naive Bayes achieved remarkable results in text classification tasks—a substantial portion of our dataset.
Execution Speed: In our operational environment, speed is a crucial aspect. When processing huge datasets, Naive Bayes is renowned for its effectiveness and quickness (here for predicting the bottom_category _id). The fact that naive bayes efficiently executed the code far faster than the models was essential for real-time processing requirements. Even though it usually required more computing power than Naive Bayes, logistic regression provided a competitive speed that was manageable within our infrastructure constraints.

4. MODEL TRAINING

4.1 PREPARING DATA

To guarantee the best possible model performance, a thorough data preparation process was carried out before the models were trained:

Feature Encoding: To convert non-numerical labels into a numeric format, all categorical features were encoded. For the target variable, the LabelEncoder was used to transform it into a format appropriate for model training.

Feature Vectorization: HashingVectorizer was used to vectorize text data from features, such as titles, descriptions, tags, and so on. Through the conversion of text into a fixed-size numerical representation, scalability and effective management of massive amounts of text data are maintained.

Data Splitting: Eighty percent of the data was set aside for testing, and the remaining twenty percent was divided at random into training and testing sets. This split was carried out to assess the model's performance on untested data, guaranteeing the model's generalizability.

4.2 MODEL CONFIGURATION

Naive Bayes:
Because of its probabilistic approach to managing independent features, Naive Bayes is used.
Due to the model's computational efficiency and feature independence assumption, it was specifically selected for its efficacy in text classification.

Logistic Regression:
It was configured with max_iter=1000 in

order to guarantee convergence in light of our high-dimensional data's complexity. In order to guarantee reproducibility of results, random_state=42 was specified. To expedite the procedure and stop data leaking, each model was wrapped in a pipeline that comprises preprocessing processes (vectorization and TF-IDF transformation).

5. EVALUATION

5.1 Predicting Top Category ID

Two methods were tested in the first part of our project: Random Forest and Logistic Regression, for predicting the 'Top Category ID'.

With an F1 score of 0.80, Random Forest demonstrated its resilience and capacity to manage intricate nonlinear interactions in the data. But logistic regression, which produced a higher F1 score of 0.89, fared better than it. Logistic Regression was used for the final predictions on the test dataset for 'Top Category ID' because of its better performance and efficiency in managing linear relationships.

5.2 Predicting Bottom Category ID

Both Linear SVM and Naive Bayes were assessed for 'Bottom Category ID' prediction:

An F1 score of 0.58 and a Macro F1 score of 0.579 were obtained using linear SVM. Longer run times were caused by the model's high computational load, even with its moderate accuracy.
Although Naive Bayes produced a somewhat lower Macro F1 score of 0.535, its execution speed was noticeably faster. Naive Bayes was selected for the final 'Bottom Category ID' predictions due to the practical requirement for faster response times.

5.3 Predicting Primary Color ID

'Primary Color ID' prediction was made with the help of logistic regression and linear naive bayes:

The Macro F1 score for both models was 0.451. The final prediction was made using Naive Bayes because to its computational efficiency, which offers faster processing times that are essential for scalable deployments, even though the accuracy performance was the same.

5.4 Predicting Secondary Color ID

The last category, "Secondary Color ID," was examined using Logistic Regression, Naive Bayes, and Linear SVM:

Although not very high, Linear SVM's Macro F1 score of 0.278 made it the best model tested for this category. Consequently, in spite of its reduced efficiency, it was employed for the final forecasts.
In terms of accuracy, Naive Bayes and Logistic Regression performed worse than Linear SVM, producing scores of 0.127 and 0.25, respectively.

5.5 Justification for Model Selection

A trade-off between computing efficiency and predictive accuracy impacted which particular models were chosen for each prediction task. A common preference was for Logistic Regression because of its excellent accuracy and low computational requirements. Naive Bayes was selected despite minor accuracy losses for situations when speed was of the essence. Despite being slower, linear SVM was chosen because it performed the best in one of the scenarios, demonstrating how our model strategy trades accuracy for speed.

6. CONCLUSION

With a particular focus on predicting "Top Category ID," "Bottom Category ID," "Primary Color ID," and "Secondary Color ID," the goal of this project was to create

predictive models for Etsy's classification tasks. After a thorough analysis, we decided that Logistic Regression should be the top category ID because of its higher F1 score of 0.89, which outperforms Random Forest in terms of efficiency and accuracy. Naive Bayes was selected for the bottom category ID due to its quicker performance, which is crucial for real-time processing, even though Linear SVM had a little greater accuracy. Similarly, Naive Bayes was preferred over Logistic Regression once more for the primary color ID. Both models achieved a Macro F1 score of 0.451, with the former offering faster computing speeds. In the instance of the secondary color ID, Linear SVM was chosen due to its greatest accuracy even though it was computationally costly; this illustrates the essential trade-off between prediction precision and speed in complicated classifications.

The project's results demonstrate how crucial it is to strike a balance between computing efficiency and accuracy in order to meet the needs of an online marketplace like Etsy. The implementation of these models yields valuable insights that can be utilized to refine future forecasting strategies and operational enhancements. In the future, our predictive capabilities will adapt to changing market dynamics and user needs thanks to the integration of more advanced modeling techniques and ongoing model adjustments based on incoming data. This will support Etsy's ongoing efforts to optimize their platform and user experience.

7. BIBLIOGRAPHY

Fang, Y., Wang, J., Jia, L. and Kin, F.W., 2021, September. Shopee price match guarantee algorithm based on multimodal learning. In 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI) (pp. 59-62). IEEE.

Bhavani, A. and Santhosh Kumar, B. (2021). A Review of State Art of Text Classification Algorithms. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). doi:https://doi.org/10.1109/iccmc51019.2021.9418262.

Khurana, U., Samulowitz, H. and Turaga, D., 2018, April. Feature engineering for predictive modeling using reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

Luo, X., 2021. Efficient English text classification using selected machine learning techniques. Alexandria Engineering Journal, 60(3), pp.3401-3409.

Sinha, A., Naskar, M.N.B., Pandey, M. and Rautaray, S.S., 2022, December. Text classification using machine learning techniques: Comparative analysis. In 2022 OITS International Conference on Information Technology (OCIT) (pp. 102-107). IEEE.

Chandrashekar, G. and Sahin, F., 2014. A survey on feature selection methods. Computers & electrical engineering, 40(1), pp.16-28.

Xu, S., Li, Y. and Wang, Z., 2017. Bayesian multinomial Naïve Bayes classifier to text classification. In Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2017 11 (pp. 347-352). Springer Singapore.

Aggarwal, C.C. and Zhai, C., 2012. A survey of text classification algorithms. Mining text data, pp.163-222.

Uysal, A.K. and Gunal, S., 2014. The impact of preprocessing on text classification. Information processing & management, 50(1), pp.104-112.

Mohammad, A.H., Alwada ‘n, T. and Al-Momani, O., 2016. Arabic text categorization using support vector machine, Naïve Bayes and neural network. GSTF Journal on Computing (JoC), 5, pp.1-8.